Diese Website verwendet Cookies von Google, um Dienste anzubieten und den Traffic zu analysieren. Deine IP-Adresse und dein User-Agent werden zusammen mit Messwerten zur Leistung und Sicherheit für Google freigegeben. So können Nutzungsstatistiken generiert, Missbrauchsfälle erkannt und behoben und die Qualität des Dienstes gewährleistet werden.

WEITERE INFORMATIONEN OK

Thursday, December 11, 2014

The dawn of trustworthy computing

When we currently use a smart phone or a laptop on a cell network or the Internet, the other end of these interactions typically run on other solo computers, such as web servers. Practically all of these machines have architectures that were designed to be controlled by a single person or a hierarchy of people who know and trust each other. From the point of view of a remote web or app user, these architectures are based on full trust in an unknown "root" administrator, who can control everything that happens on the server: they can read, alter, delete, or block any data on that computer at will. Even data sent encrypted over a network is eventually unencrypted and ends up on a computer controlled in this total way. With current web services we are fully trusting, in other words we are fully vulnerable to, the computer, or more specifically the people who have access to that computer, both insiders and hackers, to faithfully execute our orders, secure our payments, and so on. If somebody on the other end wants to ignore or falsify what you've instructed the web server to do, no strong security is stopping them, only fallible and expensive human institutions which often stop at national borders.

The high vulnerability we have to web servers stands in sharp contrast to traditional commercial protocols, such as ticket-selling at a movie theater, that distribute a transaction so that no employee can steal money or resources undetected. There is no "root administrator" at a movie theater who can pocket your cash undetected. Because, unlike a web server, these traditional protocols, called financial controls, can securely handle cash, you didn't have to fill out a form to see a movie, shop for groceries, or conduct most other kinds of every-day commerce. You just plunked down some coin and took your stuff or your seat. Imperfect and slow as these processes often are (or were), these analog or paper-based institutions often provided security, financial control, and/or verifiability of fiduciary transactions in many ways far superior to what is possible on web servers, at much less hassle and privacy loss to customers. On the Internet, instead of securely and reliably handing over cash and getting our goods or services, or at least a ticket, we have to fill out forms and make ourselves vulnerable to identity theft in order to participate in e-commerce, and it often is very difficult to prohibitive to conduct many kinds of commerce, even purely online kinds, across borders and other trust boundaries. Today's computers are not very trustworthy, but they are so astronomically faster than humans at so many important tasks that we use them heavily anyway. We reap the tremendous benefits of computers and public networks at large costs of identity fraud and other increasingly disastrous attacks.

Recently developed and developing technology, often called "the block chain", is starting to change this. A block chain computer is a virtual computer, a computer in the cloud, shared across many traditional computers and protected by cryptography and consensus technology. A Turing-complete block chain with large state gives us this shared computer. Earlier efforts included state-machine replication (see list of papers linked below). QuixCoin is a recent and Ethereum is a current project that has implemented such a scheme. These block chain computers will allow us to put the most crucial parts of our online protocols on a far more reliable and

About Me

Nick Szabo

"A premier thinker about history, law and economics, and the lessons they have for security." -- Adam Shostack, Emergent Chaos

"Szabo comes out with these essays that leave me in awe." -- Brian Dunbar

"Reading material that is eclectic, challenging and endlessly fascinating." -- Sean McGrath, Propylon

"Like most blogs worth my attention, this blog is updated only infrequently. That is because the authors of blogs worth my attention only post when they have something to say that is true, relevant and not already known by their audience. Most of the human race does not have the skill to know when an idea has these three properties. The skill is particularly rare in the fields of politics and economics, which is why this blog is such a rare and valuable thing." -- Richard Hollerith

secure footing, and make possible fiduciary interactions that we previously dared not do on a global network

Much as pocket calculators pioneered an early era of limited personal computing before the dawn of the general-purpose personal computer, Bitcoin has pioneered the field of trustworthy computing with a partial block chain computer. Bitcoin has implemented a currency in which someone in Zimbabwe can pay somebody in Albania without any dependence on local institutions, and can do a number of other interesting trust-minimized operations, including multiple signature authority. But the limits of Bitcoin's language and its tiny memory mean it can't be used for most other fiduciary applications, the most obvious example being risk pools that share collateral across a pool of financial instruments.

A block-chain computer, in sharp contrast to a web server, is shared across many such traditional computers controlled by dozens to thousands of people. By its very design each computer checks each other's work, and thus a block chain computer reliably and securely executes our instructions up to the security limits of block chain technology, which is known formally as anonymous and probabilistic Byzantine consensus (sometimes also called Nakamoto consensus). The most famous security limit is the much-discussed "51% attack". We won't discuss this limit the underlying technology further here, other than saying that the oft-used word "trustless" is exaggerated shorthand for the more accurate mouthful "trust-minimized", which I will use here. "Trust" used in this context means the need to trust remote strangers, and thus be vulnerable to them.

Trust-minimized code means you can trust the code without trusting the owners of any particular remote computer. A smart phone user in Albania can use the block chain to interact with a computer controlled by somebody in Zimbabwe, and they don't have to know or trust each other in any way, nor do they need to depend on the institutions of either's countries, for the underlying block chain computer to run its code securely and reliably. Regardless of where any of the computers or their owners are, the block chain computer they share will execute as reliably and securely as consensus technology allows, up to the aforementioned limits. This is an extremely high level of reliability, and a very high level of security, compared to web server technology.

Instead of the cashier and ticket-ripper of the movie theater, the block chain consists of thousands of computers that can process digital tickets, money, and many other fiduciary objects in digital form. Think of thousands of robots wearing green eye shades, all checking each other's accounting. Individually the robots (or their owners) are not very trustworthy, but collectively, coordinated by mathematics, they produce results of high reliability and security.

Often block chain proponents talk about the "decentralized" block chain versus the "centralized" web or centralized institutions. It's actually the protocol (Nakamoto consensus, which is highly distributed) combined with strong cryptography, rather than just decentralization *per se*, that is the source of the far higher reliability and and much lower vulnerability of block chains. The cryptography provides an unforgeable chain of evidence for all transactions and other data uploaded to the block chain. Many other decentralized or peer-to-peer (P2P) technologies do not provide anything close to the security and reliability provided by a block chain protected by full Byzantine or Nakamoto consensus and cryptographic hash chains, but deceptively style themselves as block chains or cryptocurrency.

A big drawback is that our online and distributed block chain computer is much slower and more costly than a web server: by one very rough estimate, about 10,000 times slower and more costly, or about the same as it cost to run a program on a normal computer in 1985. For this reason, we only run on the block chain that portion of an application that needs to be the most reliable and secure: what I call fiduciary code. Since the costs of human ("wet") problems caused by the unreliability and insecurity of web servers running fiduciary code are often far higher than the extra hardware needed to run block chain code, when web server reliability and security falls short, as it often does for fiduciary computations such as payments and financial contracts, it will often make more sense to run that code on the block chain than to run it less reliably and securely on a web server. Even better, the block chain makes possible new fiduciary-intensive applications, such as posting raw money itself to the Internet, securely and reliably accessible anywhere on the globe - apps that we would never dare do with a web server.

Blog Archive

- ► 2018 (1)
- ► 2017 (3)
- ► 2016 (4)
- ► 2015 (3)
- ▼ 2014 (3)
  - ▼ December (1)
    - The dawn of trustworthy computing
  - ► October (1)
  - ► July (1)
- ► 2013 (3)
- ► 2012 (8)
- ► 2011 (12)
- ► 2010 (9)
- ► 2009 (29)
- ► 2008 (55)
- ► 2007 (47)
- ► 2006 (130)
- ► 2005 (44)

What kinds of fiduciary code can we run? We are still thinking up new applications and the categories will be in flux, but a very productive approach is to think of fiduciary applications by analogy to traditional legal code that governs traditional fiduciary institutions. Fiduciary code will often execute some of the functions traditionally thought of as the role of commercial law or security, but with software that securely and reliably spans the global regardless of traditional jurisdiction. Thus:

* Property titles (registered assets), where the on-chain registry is either the legally official registry for off-chain assets or controls on-chain ones, thus providing reliable and secure custody of them. One can think of a cryptocurrency such as Bitcoin as property titles (or at least custody enforced by the block chain consensus protocol) to bits recognized as being a fixed portion of a currency, or as controlling unforgeably costly bits, or both. Block chains could also control hardware which controls the function of and access to physical property.

* Smart contracts: here users (typically two of them) agree via user interface to execute block chain code, which may include transfer of money and other chain-titled assets at various times or under various conditions, transfer and verification of other kinds of information, and other combinations of wet or traditional (off-chain) and dry (on-chain) performance. A block chain can hold cryptocurrency as collateral (like an escrow) which incentivizes off-chain performance that can be verified on-chain, by the parties or by third parties. A full block chain computer can pool on-chain assets into a single chain-controlled risk pool spread among many similar financial contracts, reducing the amount of collateral that needs to be stored on-chain while minimizing the need for off-chain collateral calls. The block chain can also make the search, negotiation, and verification phases of contracting more reliable and secure. With on-chain smart contracts we will be able to buy and sell many online services and financial instruments by button and slider instead of by laboriously filling out forms that disclose our private information.

* On-chain treasuries, trusts, and similar, where money lives on the block chain and is controlled by multiple signature ("multisig") authority. Putting a treasury with signature authority on a block chain computer is low-hanging fruit, but is often tied to more speculative efforts under the label "distributed autonomous organization (DAO)", which may include voting shares and other mechanisms to control the treasury like a corporation or other kind of of organization.

I hope to discuss these block chain applications, especially smart contracts, in future posts. While there is much futurism in many block chain discussions, including many trying to solve problems that aren't actually solved by the block chain, I will generally stick to low-hanging fruit that could be usefully implemented on Quixcoin, Ethereum, or similar technology in the near future, often interfacing to still necessary parts of traditional protocols and institutions rather than trying to reinvent and replace them in whole.

References

Here is a list of basic computer science papers describing the technology of block chains (including cryptocurrencies).

Wet vs. dry code

Posted by Nick Szabo at 10:16 AM

26 comments:

Anonymous said...

Hi Nick,

I really enjoy your work, particularly your essays on philosophy. I have to wonder in regards to this post, doesn't this mode of trustworthy computing challenge the general power structures that computer networks are built upon?

>With current web services we are fully trusting, in other words we are fully vulnerable to, the computer, or more specifically the people who have access to that computer, both insiders and hackers, to faithfully execute our orders, secure our payments, and so on. If somebody on the other end wants to ignore or falsify what you've instructed the web server to do, no strong security is stopping them, only fallible and expensive human institutions which often stop at national borders.

It seems to me that this flips the law as it stands on its head. No longer is power enumerated into gatekeepers, but to the gates themselves.

Anyways, I love your thoughtful work and look forward to seeing what you come up with next.

10:45 AM

Anonymous said...

"deveptively" I think you meant "deceptively"

10:57 AM

Anonymous said...

Double space after the first use of "Nakamoto"! hmm, fishy...

11:33 AM

Richard said...

in case the other post doesn't post is the basic take away that accountability is becoming a commodity at least should anyone know how to decipher that which is indicative of the actions requiring or creating accountability?

12:16 PM

Anonymous said...

Wonderful essay.
Does this change how you feel about futarchy?
https://github.com/psztorc/Truthcoin

12:46 PM

Tim Pastoor said...

Another great essay, and I specifically like the elaboration on the misconception between "decentralized" vs "Nakamoto consensus". Still, centralized authorities do cause a lot of harm to societies in many cases, and in most of those cases decentralization is a better option imo.

Nakamoto consensus is preferable, but only applicable to cases wherein everyone has to agree on one specific thing. If that thing is subjective to each party involved, thus highly relative and can't be solved by means of Nakamoto consensus (i.e., trust among

humans), it's not the case.

In that case, decentralized (or even distributed) peer-to-peer networks can still offer better solutions to the problems that are being caused by centralized authorities. Still, making sure that only the owner has access to his/her property remains essential, at any given moment, through strong cryptography.

One solution to solving subjective trust among individual parties can be the whitelisting principle (i.e., how humans tend to trust each other), which is of course still less secure than Nakamoto consensus. Since Nakamoto consensus can not be applied in these matters, whitelisting solutions are probably the best-effort to follow.

1:18 PM

Anonymous said...

I love reading your blogs Nick.
Stay up baby!

1:51 PM

Qointum said...

Hi Nick,

You might be interested in checking out our cryptocurrency platform, Qointum. We have smart contracts scripted in Python that are far more flexible and natural looking than what other platforms provide. Other platforms ended up with contorted designs because they entirely missed the concept of conserved transferable asset objects and code access specifiers. Other platforms are also missing a natural language for multi-chain smart contracts which allow us to scale up across the web.

You mentioned the cost to run these blockchain computers is high, not so with Qointum. We are the first to have a new proof-of-stake algorithm (rather than proof-of-work mining which wastes electricity) that scales from a single stake holder to thousands, while also allowing consensus from block headers only (without examining transactions), which lets us go multi-chain using Simplified State Verification (SSV).

2:40 PM

null said...

Another great read from Nick Szabo, or is it Satoshi Nakamoto?

3:04 PM

Mark said...

This comment has been removed by the author.

3:23 PM

Anonymous said...

Reminds me of this. http://www.loper-os.org/?p=1490

5:05 PM

Anonymous said...

What do you think of Codius (codius.org)? -- seems to address the cost and speed drawbacks of the blockchain's set amount of redundancy

5:23 PM

Jeff McDonald said...

This comment has been removed by the author.

8:41 PM

Anonymous said...

>Block chains could also control hardware which controls the function of and access to physical property.

Imagine a blockchain that operated a printing press. This press could print codes on a special piece of paper. We could trade these pieces of paper to each other in return for goods and services.

10:26 PM

Benjamin Wright said...

Nick: I humbly argue that effective "fiduciary code" and "smart contracts" will benefit from traditional legal analysis and legal draftsmanship. A skilled legal draftsman composes a better contract than someone who lacks the skill. I explain: http://hack-igations.blogspot.com/2014/11/ethereum.html

9:20 AM

Alberth said...

Could a Bitcoin hard fork solve this, eventually?

10:40 AM

Anonymous said...

You sort of lost me at the movie theater example. You clearly never worked a retail job because employees steal stuff all the time. There's plenty of ways for insiders to steal credit card numbers or cash. And on the other end of things there's just straight up embezzling by the "root" accountants or managers.

12:29 PM

Anonymous said...

"You sort of lost me at the movie theater example. You clearly never worked a retail job because employees steal stuff all the time. There's plenty of ways for insiders to steal credit card numbers or cash. And on the other end of things there's just straight up embezzling by the "root" accountants or managers."

I think the main point is not so much that. It's about the difference between a physical and a digital transaction. A physical/cash transaction is "secure" not in the sense that theft cannot happen -- but rather that the exchange is defined by physical reality. You hand over cash to the deli, theatre,food truck... and that's it.

The cash left you undeniably. And the vendor knows that he received payment (I mean it's right there in the register, looks and feels like a real $20). And that's why you never once needed to fill out a form to do any of that. The cashier never had to know who the heck you were. The transaction was "secure".

In the digital world, there is much more that needs to happen behind the scenes - because of the nature of what data is -- being digital -- itself. Even the swipe of a credit card involves an initial sign-up with Identification check and it goes through a complex processes of other checks and clearances, through various gateways and systems.

The point is that there is no physical-like- "cash" equivalent that has existed. At least until now.

2:52 PM

Donald McIntyre said...

This comment has been removed by the author.

12:49 PM

Anonymous said...

"You sort of lost me at the movie theater example. You clearly never worked a retail job because employees steal stuff all the time. There's plenty of ways for insiders to steal credit card numbers or cash. And on the other end of things there's just straight up embezzling by the >>root<< accountants or managers."

If i'm not mistaken Nick meant that in the theater you have your cash and assets at all times in your pockets, administrators won't see them unless you'll expose them willingly.

The analogy of "root" administrator in a movie theater would be a deposit where you're required to store all your posessions before entering the projection room. Then the administrators would be free to copy, steal and destroy your posessions without your knowledge and there would be no way of stopping them.

Anonymously,
Przemysław Bagiński

8:47 AM

dominiksf said...

funny congruency in thought, i posted this around the same time. great to see others thinking in this direction.

https://bitcointalk.org/index.php?topic=400235.msg9795832#msg9795832

4:52 PM

Anonymous said...

Hi Nick,

As always great work. I really appreciate the knowledge and information you make available to the common person like myself that doesn't always have easy access to highly guarded research and study findings.

Also, great job on finally putting a name to some of the stuff we are dealing with on a daily basis. The "Virtual Computer" Blockchain reference is great.

Respectfully,
L. Detweiler

4:48 AM

David Gerard said...

In your original 1994 paper on smart contracts, you note the necessity of human intervention in case something goes wrong.

It's entirely unclear to me how this is possible with a blockchain enforced by strong cryptography.

The plot of Dr Strangelove is literally a smart contract going wrong and people's attempts to avert the preprogrammed outcome.

Reddit post: http://www.reddit.com/r/Buttcoin/comments/2pzssy/in_what_world_are_smart_contracts_actually_a_good/

2:12 PM

Chris said...

In the context of the movie theater and grocery store example, I was about to say that Nick has been thinking about this for a long time. Then I googled it, and found this article from 1997.

The point is that at the movie theater, there's a paper trail showing how much money was received, and the cashier can't steal untraceably from the owners. Customers insist on getting a ticket in exchange for their money. And the ticket taker is separate to ensure that cashiers don't let their friends in for half price.

At the grocery store, similarly, there's a cash register tracking every transaction. The owner encourages the customers to insist on receipts ("If I forget to give you a receipt, we'll refund half your money.")

Institutions like these don't prevent someone from opening the till and running away with the funds, but they do make it evident if someone tries to skim 1% of the money over the long term.

10:54 PM

Anonymous said...

Post by eddie:

Although smart contracts are still in their nascent stage, the potential is clear. If a simple enough user interface were developed it could remove a host of legal headaches, like updating your will. Imagine if allocating your assets after your death was as simple as moving an adjustable slider that determines who gets how much. Once the smart contract can verify the triggering condition--in this case, your death--the contract goes into effect and your assets are divvied up.

A smart contract can be activated, after which point it takes on a "life of its own", in the sense that it will process regularly over time according to its own internal terms and logic, until it expires or is deactivated. In this sense, it is like any other recurring OT transaction such as market trades and payment plans.

Smart contracts are most distinguished by the fact that they can have scriptable clauses. (Normally on OT, a typical contract only

contains data tags, and human-readable clauses. Only the smart contracts "can come to life" with their own custom-scripted clauses.)

Smart contracts can have multiple parties, each with their own agents and asset accounts.

Only a select few functions are made available from within the script code of a smart contract, and all funds transfers go through this tightly-controlled interface, with notices and receipts sent to all relevant parties.

The script code is unable to manipulate any assets excepting those explicitly declared beforehand on the smart contract, and verified as valid property of a legitimate party to the contract. And when funds are moved, it's to a contract-specific name, not an account ID. For example, you wouldn't transfer funds from account pckjsdf9872345kj34kjhsf, but rather, by that account's name as declared in the contract, such as alice_petty_cash or bobs_acct. (This means that it's impossible to even reference any account other than those declared on the smart contract by name, since the functions provided can only operate based on those names.)

1:46 PM

Anonymous said...

Please write more. I'm thirsty for more of this original stuff.

1:14 PM

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Simple theme. Powered by Blogger.